# Telco Data Check & Analyze

- Write a C++ program to perform some queries on a telco data (comming from stdin) with the following format:
- The first block of data consists of lines (terminated by a line containing #), each line (number of lines can be up to 100000) is under the form:

  call <from_number> <to_number> <date> <from_time> <end_time>

  - which is a call from the phone number <from_number> to a phone number <to_number> on <date>, and starting at time-point <from_time>, terminating at time-point <end_time>
  - <from_number> and <to_number> are string of 10 characters (a phone number is correct if it contains only digits 0,1,...,9, otherwise, the phone number is incorrect)
  - <date> is under the form YYYY-MM-DD (for example 2022-10-21)
  - <from_time> and <to_time> are under the form hh:mm:ss (for example, 10:07:23)
- The second block consists of queries (terminated by a line containing #), each query in a line (number of lines can be up to 100000) and belongs to one of the following types:
  - ?check_phone_number: print to stdout (in a new line) value 1 if no phone number is incorrect
  - ?number_calls_from <phone_number>: print to stdout (in a new line) the number of times a call is made from <phone_number>
  - ?number_total_calls: print to stdout (in a new line) the total number of calls of the data
  - ?count_time_calls_from <phone_number>: print to stdout (in a new line) the total time duration (in seconds) the calls are made from <phone_number>

# Telco Data Check & Analyze

- Example

| stdin | stdout |
| --- | --- |
| call 0912345678 0132465789 2022-07-12 10:30:23<br>10:32:00<br>call 0912345678 0945324545 2022-07-13 11:30:10<br>11:35:11<br>call 0132465789 0945324545 2022-07-13 11:30:23<br>11:32:23<br>call 0945324545 0912345678 2022-07-13 07:30:23<br>07:48:30<br>#<br>?check_phone_number<br>?number_calls_from 0912345678<br>?number_total_calls<br>?count_time_calls_from 0912345678<br>?count_time_calls_from 0132465789<br># | 1<br>2<br>4<br>398<br>120 |